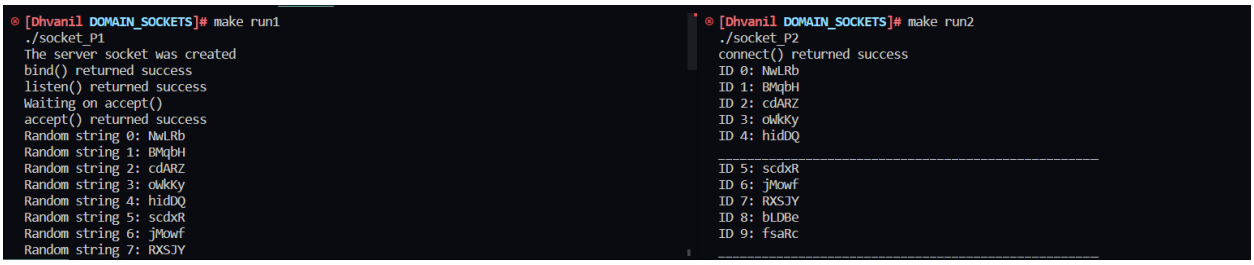
Q2 Inter-process Communication

(I) Unix Domain Sockets

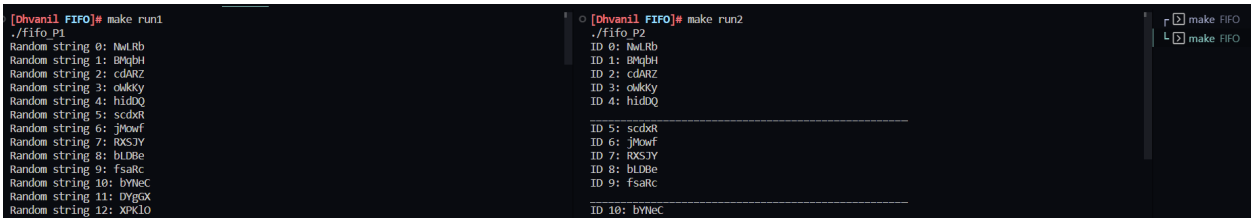
This code is for a server that sends a series of strings to a client over a Unix domain socket. The server generates 50 random strings and sends them in groups of 5 to the client. After each batch is sent, the server waits for 3 seconds and reads an acknowledgement ID from the client. The acknowledgement ID is the index of the next string to send. The server continues sending strings until it receives an acknowledgement ID of 49, indicating that all 50 strings have been sent. The server creates a Unix domain socket and listens for connections from a client. When a client connects, the server sends the strings and reads the acknowledgement ID. It then closes the socket when the process is finished. The '**write**' and '**read**' functions are used to send and receive data through the socket, and the '**close**' function is used to close the socket.



(II) FIFO's

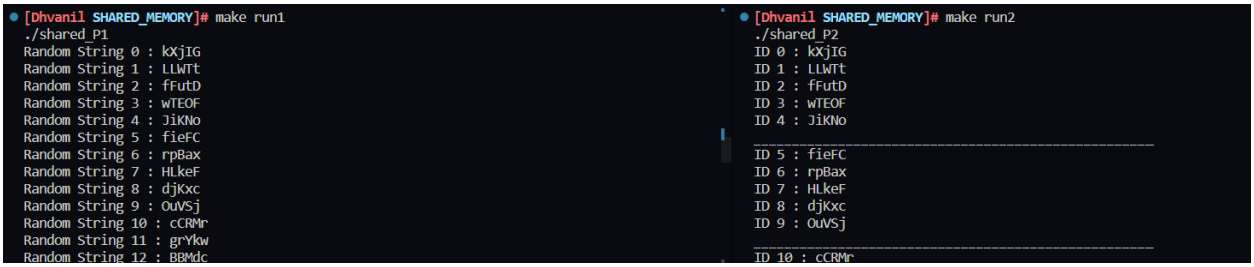
50 random strings are sent in batches of 5 to a named pipe (FIFO). It then waits for 3 seconds and reads an acknowledgement ID from the named pipe. The acknowledgement ID is the index of the next string to send. The loop ends when the acknowledgement ID is 49, indicating that all 50 strings have been sent. The second program receives a batch of 5 strings from the named pipe, processes them, and sends an acknowledgement ID back to the first program. The

acknowledgement ID is the index of the next string to receive. The loop ends when the acknowledgement ID received is 49, indicating that all 50 strings have been received. The '**write'** and '**read'** functions are used to send and receive data through the named pipe, and the '**close'** function is used to close the named pipe.



(III) **SHARED MEMORY SEGMENT**

50 random strings are sent in batches of 5 to a shared memory segment. It then waits for 3 seconds and reads an acknowledgement ID from the shared memory segment. The acknowledgement ID is the index of the next string to send. The loop ends when the acknowledgement ID is 49, indicating that all 50 strings have been sent. The second program receives a batch of 5 strings from the shared memory segment, processes them, and sends an acknowledgement ID back to the first program. The acknowledgement ID is the index of the next string to receive. The loop ends when the acknowledgement ID received is 49, indicating that all 50 strings have been received. The shared memory segment is created and accessed using the '**shmget'** and '**shmat'** functions, and strings are copied using the '**strcpy'** and '**sprintf'** functions. The '**atoi'** function is used to convert strings to integers. The current time is obtained using the 'clock_**gettime'** function.



Time Calculation Example: